MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 15, 1995	Final Report June 1992-June 1995

4. TITLE AND SUBTITLE
Computation and Implementation of Non-Monotonic Deductive Databases

5. FUNDING NUMBERS
DAAL-03-92-G-0225

6. AUTHOR(S)
V.S. Subrahmanian

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Maryland

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This is the final report on a 3-year grant from the Army Research Office to the University of Maryland to conduct research in the area of "Computation and Implementation of Non-Monotonic Deductive Databases." The main goal of the research was to study methods of efficient, scalable reasoning about situations where only incomplete and/or uncertain information is available. Towards this end, we conducted research on databases that contain incomplete symbolic information-such databases are typically called "nonmonotonic deductive databases." We also studied systems that contain incomplete information, but where different probability measures are available specifying partial information about the missing information.

15. NUMBER OF PAGES	
16. PRICE CODE	

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Final Project Report
July 15, 1995

Computation and Implementation of Nonmonotonic Deductive Databases

ARO Grant No. DAAL-03-92-G-0225

Principal Investigator: V.S.Subrahmanian
Department of Computer Science
University of Maryland
College Park, MD 20742.

This is the final report on a 3-year grant from the Army Research Office to the University of Maryland to conduct research in the area of "Computation and Implementation of Nonmonotonic Deductive Databases." The main goal of the research was to study methods of efficient, scalable reasoning about situations where only incomplete and/or uncertain information is available. Towards this end, we conducted research on databases that contain incomplete symbolic information – such databases are typically called *nonmonotonic deductive databases.* We also studied systems that contain incomplete information, but where different probability measures are available specifying partial information about the missing information.

We are pleased to report that this grant has led to 49 publications, including ones in prestigious archival journals such as The Journal of the ACM, ACM Transactions on Database Systems, IEEE Transactions on Knowledge and Data Engineering, Information and Computation, and many others.

1 Summary of Accomplishments

The research performed under this grant has been on a variety of topics related to the computation and implementation of nonmonotonic deductive databases and reasoning with heterogeneous databases, data structures and software. The research performed under this grant during this period may be classified into the following categories:

1. **Computing Minimal Models by Integer Programming.** Existing and past generations of Prolog compilers have left deduction to run-time and this may account for the poor run-time performance of existing Prolog systems. Our work tries to minimize run-time deduction by shifting the deductive process to compile-time, not run-time. Additionally, we offer an alternative inferencing procedure based on translating logic to mixed integer programming. This makes available for research and implementation in deductive databases, all the theorems, algorithms, and software packages developed by the operations research community over the past fifty years. The method keeps the same query language as for disjunctive deductive databases, only the inferencing procedure

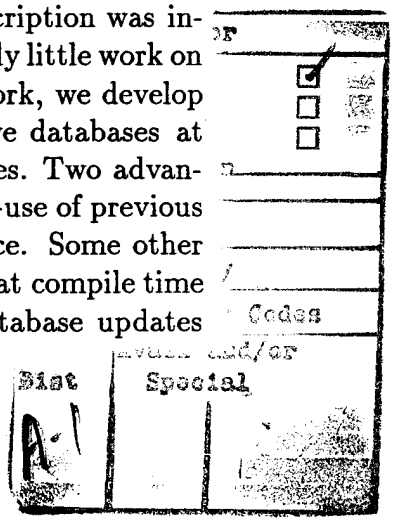
changes. The language is purely declarative, independent of the order of rules in the program, independent of the order in which literals occur in clause bodies. The technique avoids Prolog's problem of infinite looping. It saves run time by doing primary inferencing at compile time. Furthermore, it is incremental in nature.

2. **Computing Stable Models by Integer Programming.** Though the declarative semantics of both explicit and nonmonotonic negation in logic programs has been studied extensively, relatively little work has been done on computation and implementation of these semantics. In this work, we study three different approaches to computing stable models of logic programs based on mixed integer linear programming methods for automated deduction introduced by R. Jeroslow. We subsequently study he relative efficiency of these algorithms. The results of experiments with a prototype compiler implemented by us tend to confirm our theoretical discussion. In contrast to resolution, the mixed integer programming methodology is both fully declarative and handles reuse of old computations gracefully.

   We also introduce, compare, implement, and experiment with linear constraints corresponding to four semantics for "explicit" negation in logic programs: the four-valued annotated semantics of Blair and Subrahmanian, the Gelfond-Lifschitz semantics, the over-determined model semantics of Grant and Subrahmanian and the classical logic semantics. Gelfond and Lifschitz argue for simultaneous use of two modes of negation in logic programs, "classical" and "nonmonotonic", so we give algorithms for computing "answer sets" for such logic programs too.

3. **Computing Well-Founded and Stable Models.** Though the semantics of non-monotonic logic programming has been studied extensively, relatively little work has been done on operational aspects of of these semantics. In this work, we develop techniques to compute the well-founded model of a logic program. We describe a prototype implementation and show, based on experimental results, that our technique is more efficient than the standard alternating fixpoint computation. Subsequently, we develop techniques to compute the set of all stable models of a deductive database. These techniques first compute the well-founded semantics and then use an intelligent branch and bound strategy to compute the stable models. We report on our implementation, as well as on experiments that we have conducted on the efficiency of our approach.

4. **Computing Circumscriptive Databases.** Though circumscription was introduced by McCarthy over a decade ago, there has been relatively little work on algorithms for computing circumscriptive databases. In this work, we develop algorithms to compute the preferred models of circumscriptive databases at compile-time using mixed integer linear programming techniques. Two advantages of this (bottom-up) approach are that it makes efficient re-use of previous computations and it provides much faster run-time performance. Some other advantages of using linear programming to automate deduction at compile time is that its re-optimization facilities elegantly accommodate database updates



and also that it leads to a completely declarative formulation in which ordering of rules and literals in rule bodies plays no real role. Finally, we plan to use a standard relational database system as our run-time environment; this should yield relatively fast run-time processing, and provide a more expressive query language in which aggregates and the like can be expressed easily.

5. **Partial Instantiation for Definite Programs.** Query processing in *ground* definite deductive databases is known to correspond precisely to a linear programming problem. However, the "groundedness" requirement is a huge drawback to using linear programming techniques for logic program computations because the ground version of a logic program can be very large when compared to the original logic program. Furthermore, when we move from propositional logic programs to first order logic programs, this effectively means that functions symbols may not occur in clauses. In this work, we develop a theory of "instantiate-by-need" that performs instantiations (not necessarily ground instantiations) only when needed. We prove that this method is sound and complete when computing answer substitutions for non-ground logic programs including those containing function symbols. More importantly, when taken in conjunction with Colmerauer's result that unification can be viewed as linear programming, this means that resolution with unification can be completely replaced by linear programming as an operational paradigm. Additionally, our tree construction method is not rigidly tied to the linear programming paradigm – we will show that given any method M (which some implementors may prefer) that can compute the set of atomic logical consequences of a propositional logic program, our method can use M to compute the set of all (not necessarily ground) atoms that are consequences of a first-order logic program.

6. **Partial Instantiation for Minimal Model Computations.** Unlike sets of definite Horn clauses, logic programs with disjunctions of atoms in clause heads are often interpreted in terms of minimal models. It is also well known that the minimal models of logic programs are closely related to the so-called stable models of logic programs with non-monotonic negation in clause bodies, as well as to circumscription. Methods to compute minimal models of logic programs are becoming increasingly important as an intermediate step in the computation of structures associated with nonmonotonic logic programs. However, to date, all these techniques have been restricted to the case of *propositional* logic programs which means that an ordinary disjunctive logic program must be "grounded out" prior to computation. Grounding out in this manner leads to a combinatorial explosion in the number of clauses, and hence, is unacceptable. In this work, we show how, given any method M which correctly computes the set of minimal models of a propositional logic program, we can develop a strategy to compute truth in a minimal model of a disjunctive logic program P. The novel feature of our method is that it works on an "instantiate-by-need" basis, and thus avoids unnecessary grounding.

7. **Non-Ground Stable and Well-Founded Semantics for Nonmonotonic Deductive Databases:** The declarative semantics of nonmonotonic logic programming has largely been based on propositional programs. However, the ground instantiation of a logic program may be very large, and likewise, a "ground" stable model may also be very large. We develop a non-ground semantic theory for nonmonotonic logic programming. The principal advantage of this is that stable models and well-founded models can be represented as sets of atoms, rather than as sets of ground atoms. A set S of atoms may be viewed as a (usually more) compact representation of the set of ground instances of the atoms in S. We develop generalizations of the stable and well-founded semantics of logic programs based on this non-ground interpretation. The key technical basis for our theory is that of an "anticover" of a set of substitutions. Intuitively, an anticover of a set X of substitutions is a set of substitutions, all of whom are incompatible (i.e. they share no common instance) with the substitutions in X, and such that each substitution that is incompatible with all members of X is an instance of some substitution in the anticover. We develop methods of computing anticovers, show that membership in certain anticovers called optimal anticovers is decidable, and develop special complexity results in the Datalog case.

8. **Probabilistic Databases.** Of all scientific investigations into reasoning with uncertainty and chance, probability theory is perhaps the best understood paradigm. Nevertheless, all studies conducted thus far into the semantics of quantitative logic programming (cf. van Emden, Fitting, Blair and Subrahmanian, Kifer et al have restricted themselves to non-probabilistic semantical characterizations. In this work, we take a few steps towards rectifying this situation. We define a logic programming language that is syntactically similar to annotated logics, but in which the truth values are interpreted probabilistically. A probabilistic model theory and fixpoint theory is developed for such programs. This probabilistic model theory satisfies the requirements proposed by Fenstad for a function to be called probabilistic. The logical treatment of probabilities is complicated by two facts: first, that the connectives cannot be interpreted truth functionally when truth values are regarded as probabilities; second that negation-free definite clause like sentences can be inconsistent when interpreted probabilistically. We address these issues here and propose a formalism for probabilistic reasoning in logic programming. To our knowledge, this is the first probabilistic characterization of logic programming semantics. Our work is closely related to current work of Fitting, Fagin, Halpern and Megiddo, van Emden, Kifer et al and Blair and Subrahmanian.

9. **Probabilities and Nonmonotonic Reasoning:** We study the the semantics of non-monotonic negation in probabilistic deductive databases. Based on the stable semantics for classical logic programming, we examine three natural notions of stability: stable formula functions, stable families of probabilistic interpretations and stable probabilistic models. We show that stable formula functions are minimal fixpoints of operators associated with probabilistic logic

programs. We also prove that each member in a stable family of probabilistic interpretations is a probabilistic model of the program. Then we show that stable formula functions and stable families behave as duals of each other, tying together elegantly the fixpoint and model theories for probabilistic logic programs with negation. Furthermore, since a probabilistic logic program may not necessarily have a stable family of probabilistic interpretations, we provide a stable class semantics for such programs. Finally, we investigate the notion of stable probabilistic model. We show that this notion, though natural, is too weak in the probabilistic framework.

10. **Efficient View Maintenance.** We present incremental evaluation algorithms to compute changes to materialized views in relational and deductive database systems, in response to changes (insertions, deletions, and updates) to the relations. The view definitions can be in SQL or Datalog, and may use `UNION`, negation, aggregation (e.g. `SUM`, `MIN`), linear recursion, and general recursion.

    We first present a *counting* algorithm that tracks the number of alternative derivations (`counts`) for each derived tuple in a view. The algorithm works with both set and duplicate semantics. We present the algorithm for *nonrecursive views* (with negation and aggregation), and show that the `count` for a tuple can be computed at little or no cost above the cost of deriving the tuple. The algorithm is optimal in that it computes exactly those view tuples that are inserted or deleted. Note that we store only the *number* of derivations, not the derivations themselves.

    We then present the Delete and Rederive algorithm, DRED, for incremental maintenance of recursive views (negation and aggregation are permitted). The algorithm works by first deleting a superset of the tuples that need to be deleted, and then rederiving some of them. The algorithm can also be used when the view definition is itself altered.

11. **Hybrid Knowledge Bases** Deductive databases that interact with, and are accessed by, reasoning agents in the real world (such as logic controllers in automated manufacturing, weapons guidance systems, aircraft landing systems, land-vehicle maneuvering systems, and air-traffic control systems) must have the ability to deal with multiple modes of reasoning. Specifically, the types of reasoning we are concerned with include, amongst others, reasoning about time, reasoning about quantitative relationships that may be expressed in the form of differential equations or optimization problems, and reasoning about *numeric* modes of uncertainty about the domain which the database seeks to describe. Such databases may need to handle diverse forms of data structures, and frequently they may require use of the assumption-based non-monotonic representation of knowledge.

    A hybrid knowledge base is a theoretical framework capturing all the above modes of reasoning. The theory tightly unifies the Constraint Logic Programming Scheme of Jaffar and Lassez, the Generalized Annotated Logic Programming Theory of Kifer and Subrahmanian, and the Stable Model semantics of

Gelfond and Lifschitz. New techniques are introduced which extend both the work on Annotated Logic Programming and the Stable Model semantics.

12. **Integrating Relational Databases:** Previous research on integrating multiple relational databases do not consider the problem of what to do when integrity constraints are violated by the amalgamation of relational databases, even though the individual databases participating in the amalgamation satisfy the integrity constraints. We propose three new data retrieval notions based on whether the constraint semantics is "naive", "skeptical" or makes "choices." We propose a semantics for these operations, and develop an algebra and calculus based on these operators. We prove that the algebra can be embedded within the calculus – however, the calculus is strictly more powerful than the algebra. We study various algebraic properties linking the newly defined operators together and show how these algebraic properties can be used for query optimization.

13. **Distributed Knowledge Integration:** Integrating knowledge from multiple sources is an important aspect of automated reasoning systems. In previous work, we presented a uniform declarative and operational framework, based on *annotated logics*, for amalgamating multiple knowledge bases and data structures (e.g. relational, object-oriented, spatial, and temporal structures) when these knowledge bases (possibly) contain inconsistencies, uncertainties, and nonmonotonic modes of negation. We showed that annotated logics may be used, with some modifications, to *mediate* between different knowledge bases. The multiple knowledge bases are amalgamated by embedding the individual knowledge bases into a lattice. In this work, we describe how, given a network of sites where the different databases reside, it is possible to define a distributed semantics for amalgamated knowledge bases. More importantly, we study how the mediator may be distributed across multiple sites so that when certain conditions are satisfied, network failures do not affect the end results of queries that a user may pose. We specify different ways of distributing the mediator to protect against different types of network link failures and develop alternative soundness and completeness results.

14. **Efficient Querying of Amalgamated Knowledge Bases:** Integrating knowledge from multiple sources is an important aspect of automated reasoning systems. In this work, we briefly describe an SLD-resolution based proof procedure that is sound and complete w.r.t. our declarative semantics. We will then develop an OLDT-resolution based query processing procedure, `MULTI_OLDT`, that satisfies two important properties: (1) *efficient reuse* of previous computations is achieved by maintaining a table – we describe the structure of this table and show that table operations can be efficiently executed, and (2) *approximate, interruptable query answering* is achieved, i.e. it is possible to obtain an "intermediate, approximate" answer from the `QPP` by interrupting it at any point in time during its execution. The design of the `MULTI_OLDT` procedure will include: (1) development of data structures for tabling (substitution, truth value) pairs,

and (2) the development of algorithms to incrementally and efficiently update the table.

15. **Missile Siting Probelms:** Hybrid knowledge bases are a formalism for integrating multiple representations of knowledge and data. HKBs provide a uniform framework for integrating uncertain information (as is often the case in terrain reasoning), temporal information (needed for weather effects, etc), and numeric constraint solving capabilities (for situation assessment). We show how the HKB formalism may be applied to solve the problem of placing Patriot and Hawk missile batteries in a specified terrain subject to the requirement that various existing assets be afforded maximal protection. We formalize this problem in a clear, mathematical framework, using the HKB paradigm, and show how the problem is solved. This provides a mathematically sound, as well as a practically viable, scalable solution to the important problem of missile siting.

16. **Linking Planning and Non-Monotonic Reasoning:** Jointly with carlo Zaniolo, we have shown that there is a simple connection between logic programming and planning. The main result of our work is the following: given any planning domain consisting of an initial state, and a set of operation definitions, this domain can be translated, in linear-time, to a logic program such that a given goal G is achievable in the planning domain iff a related goal $G^\star$ is true in some stable model of the logic program obtained by the translation. We show that this translation yields at least two interesting consequences: (1) methods to update databases can be used to handle surprises when executing plans (i.e. a surprise occurs when an initial plan is partly executed, but one of the resulting intermediate states differs, perhaps due to external reasons, from what is predicted). (2) rigid actions, which are actions that *must* be executed when their pre-conditions are true, can be easily accommodated within our framework as well.

17. **Temporal Databases:** In a federated database environment, different constituents of the federation may use different temporal models or physical representations for the temporal information. This work introduces a new concept, called a *temporal module*, to resolve these mismatches among the constituents. Intuitively, a temporal module hides the implementation details of a temporal relation by exposing its information only through two *windowing functions*: a function associating each time point with a set of tuples and a function linking each tuple to a set of time points. A calculus-style language is given to form queries on temporal modules.

    Temporal modules are then extended to resolve another type of mismatch among the constituents of a federation, namely, mismatch involving different time units (e.g., *month*, *week* and *day*) used in recording temporal information. Our solution to this mismatch relies on "information conversions" provided by each constituent. Specifically, a temporal module is extended to provide several "windows" to its information, each in terms of a different time unit. The first step to process a query addressed to the federation is to select suitable windows for the

underlying temporal modules. In order to do so, time units are formally defined and studied. A federated temporal database model and its query language are proposed. The query language is an extension of the calculus-style language above.

18. **Security in Deductive databases:** In this work, we develop a formal logical foundation for secure deductive databases. This logical foundation is based on an extended logic involving several modal operators. We develop two models of interaction between the user and the database called "yes-no" dialogs, and "yes-no-don't know" dialogs. Both dialog frameworks allow the database to lie to the user. We develop an algorithm for answering queries using yes-no dialogs and prove that secure query processing using yes-no dialogs is NP-complete. Consequently, the degree of computational intractability of query processing with yes-no dialogs is no worse than for ordinary databases. Furthermore, the algorithm is maximally cooperative to user in the sense that lying is resorted to only when absolutely necessary. For Horn databases, we show that secure query processing can be achieved in linear time – hence, this is no more intractable than the situation in ordinary databases. Finally, we identify necessary and sufficient conditions for the database to be able to preserve security. Similar results are also obtained for yes-no-don't know dialogs.

19. **Theorem Proving with Equality:** We show that the algorithm directly induced by the viability definition of Digricoli and Harrison does not terminate in general. As a consequence, RUE-resolution in strong form is not complete. Moreover, we show that ground query processing for *covered pure logic programs* can be reduced to computing viability. Since the problem of ground query processing is strictly recursively enumerable even under the above restrictions, it follows that the notion of viability is undecidable. Finally, we present a modified viability check that solves the non-termination problem for ground terms.

2 Technology Transfer: Applications to the Army's Missile Siting Problems.

Jointly with John Benton of the US Army Topographic and Engineering Center, Corps. of Engineers, Ft. Belvoir, VA, we studied the use of deductive databases as an efficient way of siting Patriot and Hawk Missile Batteries. In a war-time situation, Patriot and Hawk missile batteries are used to protect the front line against enemy attack. If the batteries could simply be positioned and left in place for the duration of the war, then the task of manually siting these batteries would not be onerous. Unfortunately, this is normally not the case during a shooting war. On a battlefront, these weapons systems may be moved several times in one week and there must be several contingency sites available for each battery. In particular, Hawk batteries have strict requirements for locating sites with optimum line-of-sight visibility, and even

with current computer-assisted planning systems, siting the batteries is a trial-and-error based, time-consuming task. Patriot Missile Batteries can be used to defend against either air-breathing threats such as jet aircraft or cruise missiles or against ballistic missile threats such as the Scud missile. Deployment patterns for the two tasks are quite different. In the work to date, only the air breathing case is considered. We show that the HKB framework framework may be used to naturally represent the problem of siting Patriot and Hawk missile batteries in order to protect assets located in a given theater of operations.

Contact with Army Personnel: I are delighted to report a healthy, continuing ongoing interaction between myself and the AI group at the US Army Topographic and Engineering Center (TEC). In particular, since the beginning of this grant, I have visited TEC approximately 30-35 times, and John Benton and Ann Werkheiser of TEC have visited the University of Maryland at least on 20 occasions. Furthermore, Anne Brink of TEC has visited us at least 10 times. John Benton and I have been interacting on the applications of hybrid knowledge base technology to the problem of siting Patriot and Hawk missile batteries. Hybrid knowledge bases are a class of knowledge bases used to integrate diverse representations of data, and, in some cases, diverse reasoning paradigms.

3 Graduate Students

One PhD student (Raymond Ng) graduated while being supported, in part, by this grant. Another student received a Master's Thesis (Ross Emery). Other students who have been supported, in part, by this grant include Tom Weaver, Sibel Adali and K.S. Candan.

4 Publications

1. S. Adali, K.S. Candan, S.-S. Chen, K. Erol, and V.S.Subrahmanian. Advanced Video Information Systems. Submitted to ACM MULTIMEDIA JOURNAL, currently being revised according to referee comments.

2. V.S. Subrahmanian. Amalgamating Knowledge Bases, ACM TRANSACTIONS ON DATABASE SYSTEMS, 19, 2, pp. 291–331, 1994.

3. S. Adali and V.S. Subrahmanian. Amalgamating Knowledge Bases, II: Distributed Mediators; has appeared in: INTERNATIONAL JOURNAL OF INTELLIGENT COOPERATIVE INFORMATION SYSTEMS, vol. 3, No. 4 (1994), 349-383.

4. S. Adali and V.S. Subrahmanian. Amalgamating Knowledge Bases, III: Algorithms, Data Structures and Query Processing; UNIVERISTY OF MARYLAND TECHNICAL REPORT. Accepted for publication in Journal of Logic Programming.

5. Kasim S. Candan, V.S. Subrahmanian. An Algebra and Calculus for Multidatabases with Integrity Constraints; UNIVERSITY OF MARYLAND TECHNICAL REPORT.

6. S. Pradhan, J. Minker and V.S. Subrahmanian. Combining Databases with Prioritized Information, accepted for publication in: JOURNAL OF INTELLIGENT INFORMATION SYSTEMS.

7. C. Baral, S. Kraus, J. Minker, V.S. Subrahmanian. Combining Default Logic Databases. To appear in *Journal of Intelligent Information Systems, 1995.*

8. J. Lu and V.S. Subrahmanian. Completeness Issues in RUE–NRF Deduction, JOURNAL OF AUTOMATED REASONING, 10, pps 371–388, 1993.

9. J. Minker and V.S. Subrahmanian. *Completion Theoretic Semantics for Disjunctive Logic Programs Proc. 1990 Int. Symp. on Methodologies for Intelligent Systems*, pps 545–552, Knoxville, Tennessee, North Holland. Aug. 1990.

10. K. Erol, D.S. Nau and V.S. Subrahmanian. Complexity, Decidability and Undecidability Results for Domain-Independent Planning. accepted for publication in ARTIFICIAL INTELLIGENCE journal, Special Issue on Planning and Scheduling.

11. Anil Nerode, Raymond T. Ng, V.S. Subrahmanian. Computing Circumscriptive Databases, Part 1: Theory and Algorithms. appeared in INFORMATION AND COMPUTATION, VOL. 116, NR. 1, PPS 58-80.

12. Vadim Kagan, Anil Nerode, V.S. Subrahmanian. Computing Definite Logic Programs by Partial Instantiation and Linear Programming. appeared in ANNALS OF PURE AND APPLIED LOGIC 67 (1994), PPS 161-182.

13. V. Kagan, A. Nerode and V.S. Subrahmanian. Computing Minimal Models by Partial Instantiation. accepted for publication in: THEORETICAL COMPUTER SCIENCE.

14. P. Bonatti, S. Kraus and V.S. Subrahmanian. Declarative Foundations of Secure Deductive Databases, *Proc. 1992 Intl. Conf. on Database Theory,* Lecture Notes in Computer Science, Vol. 646, pp. 391–406, Springer.

15. J. Fernandez, J. Lobo, J. Minker and V.S. Subrahmanian. Disjunctive LP + Integrity Constraints = Stable Model Semantics, ANNALS OF MATHEMATICS AND ARTIFICIAL INTELLIGENCE, Vol. 8, pps 449–474, 1993.

16. J. Lu, G. Moerkotte, J. Schue, and V.S. Subrahmanian. Efficient Maintenance of Materialized Mediated Views, in: *Proc. 1995 ACM SIGMOD Conf. on Management of Data*, San Jose, CA, May 1995.

17. J. Benton, S.S. Iyengar, W. Deng, N. Brener, and V.S. Subrahmanian. Fine-Grained Terrain Data: A New Computational Challenge for Route Planners. submitted to *Artificial Intelligence Journal.*

18. P. Bonatti, S. Kraus and V.S. Subrahmanian. Foundations of Secure Deductive Databases, in: IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Vol. 7, no. 3, June 1995.

19. S. Marcus and V.S. Subrahmanian. Foundations of Multimedia Information Systems. submitted to The Journal of the ACM.

20. V.S. Subrahmanian, S. Adali, A. Brink, R. Emery, J. Lu, A. Rajput, T.J. Rogers, R. Ross and C. Ward. HERMES: A Heterogeneous Reasoning and Mediator System. University of Maryland Technical Report.

21. A. Brink, S. Marcus and V.S. Subrahmanian. Heterogeneous Multimedia Reasoning. Accepted for publication in: IEEE COMPUTER, SEPTEMBER 1995.

22. J. Lu, A. Nerode and V.S. Subrahmanian. Hybrid Knowledge Bases, accepted for publication in: IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

23. John Horst, Ernest Kent, Hassan Rifky, V.S. Subrahmanian. (ed. E. Gelsema and L. Kanal). Hybrid Knowledge Bases for Real-Time Robotic Reasoning, in *Proc. Pattern Recognition in Practice IV Conference, June 1994*, N. Holland, Elserver.

24. V.S. Subrahmanian. Hybrid Knowledge Bases for Intelligent Reasoning Systems, Invited Address, Proc. 8th Italian Conf. on Logic Programming, pps 3–17, Gizzeria, Italy, June 1993.

25. V.S. Subrahmanian. Hybrid Knowledge Systems, Proc. *Proc. 7th Space Operations, Applications and Research Symp.*, pps 218–225, Aug. 1993, Houston, Texas.

26. C. Bell, A. Nerode, R. Ng and V.S. Subrahmanian. Implementing Deductive Databases by Linear Programming, *Proc. 1992 ACM SIGMOD/SIGACT/SIGART Symp. on Principles of Database Systems*, pps 283–291, San Diego, May 1992.

27. C. Bell, A. Nerode, R. Ng and V.S. Subrahmanian. Implementing Deductive Databases by Mixed Integer Programming, accepted for publication in ACM TRANSACTIONS ON DATABASES SYSTEMS.

28. C. Bell, A. Nerode, R. Ng and V.S. Subrahmanian. Implementing Stable Semantics by Linear Programming, *Proc. 1993 Intl. Workshop on Logic Programming and Nonmonotonic Reasoning*, pps 23–42, Lisbon, Portugal, June 1993. MIT Press.

29. Sibel Adali and V.S. Subrahmanian. Intelligent Caching in Hybrid Knowledge Bases, in: *Proc. 1995 Intl. Conf. on Very Large Knowledge Bases*, IOS Press, Twente, The Netherlands, May 1995.

30. A. Gupta, I.S. Mumick and V.S. Subrahmanian. Maintaining Views Incrementally, *Proc. 1993 ACM SIGMOD Conf. on Management of Data*, pps 157–165, Washington, DC. (with A. Gupta and I. S. Mumick).

31. D. Perlis and V.S. Subrahmanian. Meta-languages, Reflection Principles, and Self-Reference, in: "Handbook of Logic in Artificial Intelligence and Logic Programming", pps 323–358, ed. Dov Gabbay, Oxford Univ. Press, 1994.

32. C. Bell, A. Nerode, R. Ng and V.S. Subrahmanian. Mixed Integer Programming Methods for Computing Non-Monotonic Deductive Databases, JOURNAL OF THE ACM, Vol. 41, Nr. 6, pps 1178–1215, Nov. 1994.

33. S. Marcus, V.S. Subrahmanian. Multimedia Database Systems, to appear in MULTIMEDIA DATABASE SYSTEMS: RESEARCH ISSUES AND DIRECTION, SPRINGER VERLAG.

34. R. Ng and V.S. Subrahmanian. Probabilistic Logic Programming, INFORMATION AND COMPUTATION, 101, 2, pps 150–201, 1993.

35. S. Kraus and V.S. Subrahmanian. Multiagent Reasoning with Probability, Time and Beliefs, INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, 10(5) 459–499.

36. G. Gottlob, S. Marcus, A. Nerode and V.S. Subrahmanian. Non-Ground Stable and Well-Founded Semantics. submitted to THEORETICAL COMPUTER SCIENCE journal, currently under revision.

37. K. Erol, D. Nau and V.S. Subrahmanian. On the Complexity of Domain-Independent Planning, *Proc. 1992 Conf. of the American Association for Artificial Intelligence (AAAI-92)*, pps. 381–386, MIT Press, July 1992.

38. E. Hwang and V.S. Subrahmanian. Querying Video Libraries. Accepted for publication in: JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION.

39. J. Grant and V.S. Subrahmanian. Reasoning About Inconsistent Knowledge Bases, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 7,1, pps 177–189.

40. S. Marcus and V.S. Subrahmanian. Relating Database Updates to Nonmonotonic Reasoning. UNIVERSITY OF MARYLAND TECHNICAL REPORT.

41. J. Lobo and V.S. Subrahmanian. Relating Minimal Models and Pre-Requisite-Free Normal Defaults, INFORMATION PROCESSING LETTERS, 44, pps 129–133, 1992.

42. V.S. Subrahmanian and C. Zaniolo. Relating Stable Models and AI Planning Domains, in: *Proc. 1995 Intl. Conf. on Logic Programming*, Tokyo, Japan.

43. K.S. Candan, S. Jajodia, V.S. Subrahmanian. Secure Mediated Databases; submitted for publication.

44. X. Y. Wang, S. Jajodia, and V.S. Subrahmanian. Temporal Modules: An Approach Toward Federated Temporal Databases, INFORMATION SCIENCES, Vol. 82, pps 103–128, 1995.

45. K.S. Candan, J. Grant, V.S. Subrahmanian. A Unified Treatment of Null Values. UNIVERSITY OF MARYLAND TECHNICAL REPORT; submitted to *VLDB* journal.

46. S. Adali, R. Emery. A Uniform Framework for Integrating Knowledge in Heterogenous Knowledge Systems. to appear in PROCEEDINGS 1995 INTERNATIONAL CONFERENCE IN DATA ENGINEERING, TAIPEI, TAIWAN.

47. J. Benton and V.S. Subrahmanian. Using Hybrid Knowledge Bases for Missile Siting Problems, Proc. 1994 Conf. on Artificial Intelligence Applications, pps 141–148, IEEE Computer Society.

48. V.S. Subrahmanian, D.S. Nau and C. Vago. WFS + Branch and Bound = Stable Models, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Vol. 7, no. 3, June 1995.

49. J. Benton, S.S. Iyengar, W. Deng, N. Brener, and V.S. Subrahmanian. Tactical route planning: New algorithms for decomposing the map, accepted for publication in: Proc. 1995 IEEE Intl. Conf. on Tools for Artificial Intelligence, Washington DC, Nov. 1995.